

学校编码: 10384

分类号_____密级_____

学号: X2010230460

UDC _____

厦门大学

工 程 硕 士 学 位 论 文

银行业税务发票管理系统的设计与实现

Design and Implementation of the Banking Tax Invoice
Management System

范敬巍

指 导 教 师: 王 鸿 吉 副 教 授

专 业 名 称: 软 件 工 程

论文提交日期: 2012 年 10 月

论文答辩日期: 2012 年 11 月

学位授予日期: 年 月

答辩委员会主席: _____

评 阅 人: _____

2012 年 10 月

厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下,独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果,均在文中以适当方式明确标明,并符合法律规范和《厦门大学研究生学术活动规范(试行)》。

另外,该学位论文为()课题(组)的研究成果,获得()课题(组)经费或实验室的资助,在()实验室完成。(请在以上括号内填写课题或课题组负责人或实验室名称,未有此项声明内容的,可以不作特别声明。)

声明人(签名):

2012 年 月 日

厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

（ ） 1. 经厦门大学保密委员会审查核定的保密学位论文，
于 年 月 日解密，解密后适用上述授权。

（ ☒ ） 2. 不保密，适用上述授权。

（请在以上相应括号内打“√”或填上相应内容。保密学位论文应是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

2012 年 月 日

摘 要

随着中国银行业的快速发展,经营模式及业务种类越来越多元化,银行需要开具大量的发票。现实情况显示,大多数银行目前对发票还停留在手工管理模式,巨大的开票量导致出现诸如:分行与网点间信息交互不及时、客户获取发票周期长、网点发票保管及开具无法监控、不利于开票统计等一系列的问题。从 2010 年起,国家新的《发票管理办法》出台有关银行业发票管理的规定,按照国家税务总局发票综合改革的要求,银行业代收费业务的发票开具可以采用自开票系统,但要实现与各地税务机关网上开票系统的对接,以达到发票开具信息的适时传输。

本文认真分析了银行业发票管理的现状及业务流程,以信息化为手段,建立一个高效的银行业发票管理系统,全面管控从发票购买→发票分发→发票领用→发票审核→发票填报→发票打印→发票签收→发票验旧的完整过程,并可与税务系统数据对接,起到利于监管、方便统计的作用。

系统在SOA面向服务的架构下,采用J2EE技术、SQL Server2008数据库和Struts、Spring、Hibernate等开源框架,运用了大量先进的诸如Ajax、RSS等Web2.0的技术,增强了系统的柔性和开放性。

本文介绍论文研究背景、研究目标和业务需求,着重论述系统功能的设计和实现,从系统目标、设计思路和需求分析着手,描述系统总体架构的设计方案、系统设计的关键点和关键技术,并完成了系统的主要功能模块。

关键词: 银行业; 税务发票; 管理信息系统

Abstract

Along with the Chinese banking industry's rapid development, business model and business types become more and more diversified, the bank needs to issue a receipt. The reality show, most banks present on the invoice still stay on the manual management mode, great billing amount to appear such as: branch and network information interaction between time, customers get invoice cycle long, custody of the invoices and issue the dot cannot monitor, is not conducive to the billing system gauge and a series of problems. From 2010 onwards, the new national "invoice management approach " introduced in the banking invoice management regulations, according to the State Administration of Taxation invoice comprehensive reform, banking fees on behalf of business invoices can be used from billing system, but to achieve with the local taxation authorities online billing systems, to achieve the invoice information timely transmission.

This thesis analyzed the banking invoice management situation and business process, with the information as the means, the establishment of an efficient banking invoice management system, overall control from the invoice purchase, distribution of invoice, invoice, invoice, invoice verification for reporting - > print invoices, invoice receipt invoice verification - old complete process, and can be and the tax system data docking, to benefit, convenient regulation role of statistics.

System in the SOA service oriented architecture, using J2EE technology, SQL Server2008 database and Struts, Spring, and other open source framework, using a large number of advanced such as Ajax, RSS, Web2.0 technology, enhances the system flexibility and openness.

This thesis introduces the research background, objectives and business needs, focuses on system function design and the realization, from the system target, design ideas and demand analysis, describes the overall system architecture design, system design key points and key technology, and completed the main function module of the system.

Keywords: Banking Industry; Tax Invoice; MIS

目 录

| | |
|--------------------------|----------|
| 第一章 绪论 | 1 |
| 1.1 研究背景及意义 | 1 |
| 1.2 研究目标 | 1 |
| 1.3 研究内容 | 2 |
| 1.4 论文的结构安排 | 2 |
| 第二章 相关技术介绍 | 3 |
| 2.1 软件架构技术 | 3 |
| 2.2 面向对象设计技术 | 4 |
| 2.3 软件开发技术 | 4 |
| 2.4 JAVA 技术 | 5 |
| 2.5 SQL SERVER 2008 技术 | 7 |
| 2.6 本章小结 | 7 |
| 第三章 系统需求分析 | 8 |
| 3.1 系统概述 | 8 |
| 3.1.1 规划目标 | 8 |
| 3.1.2 术语定义 | 8 |
| 3.2 业务描述 | 9 |
| 3.2.1 系统干系人 | 9 |
| 3.2.2 系统边界 | 10 |
| 3.2.3 业务流程分析 | 11 |
| 3.2.4 业务表单 | 13 |
| 3.3 数据描述 | 14 |
| 3.3.1 静态数据 | 14 |
| 3.3.2 动态数据 | 15 |
| 3.3.3 数据库介绍 | 16 |
| 3.4 功能描述 | 17 |
| 3.4.1 用例总图 | 17 |
| 3.4.2 登录用例 | 18 |
| 3.4.3 发票购买用例 | 20 |
| 3.4.4 发票领用用例 | 22 |
| 3.4.5 发票填开用例 | 24 |
| 3.4.6 领域模型(Domain Model) | 26 |

| | |
|---|-----------|
| 3.4.7 系统顺序图 (System Sequence Diagram) | 26 |
| 3.5 性能需求 | 28 |
| 3.5.1 数据精确度 | 28 |
| 3.5.2 时间特性 | 29 |
| 3.5.3 适应性 | 29 |
| 3.6 系统运行 | 29 |
| 3.6.1 环境要求 | 29 |
| 3.6.2 硬件接口 | 31 |
| 3.6.3 软件接口 | 31 |
| 3.6.4 故障处理 | 34 |
| 3.7 其他需求 | 35 |
| 3.7.1 安全策略 | 35 |
| 3.7.2 兼容性要求 | 37 |
| 3.7.3 易用性要求 | 37 |
| 3.7.4 可维性要求 | 37 |
| 3.8 本章小结 | 37 |
| 第四章 系统总体设计 | 38 |
| 4.1 系统架构设计 | 38 |
| 4.2 系统总体模块设计 | 40 |
| 4.3 数据库设计 | 42 |
| 4.3.1 设计原则 | 42 |
| 4.3.2 基础表设计 | 43 |
| 4.3.3 系统表设计 | 49 |
| 4.3.4 编码表设计 | 49 |
| 4.3.5 业务表设计 | 50 |
| 4.4 E-R 关系图 | 62 |
| 4.5 本章小结 | 64 |
| 第五章 系统详细设计与实现 | 65 |
| 5.1 配置管理 | 65 |
| 5.1.1 开发配置 | 65 |
| 5.1.2 开发工具 | 66 |
| 5.1.3 开发环境 | 67 |
| 5.2 系统管理的相关模块 | 68 |
| 5.2.1 登录管理 | 68 |
| 5.2.2 首页管理 | 70 |

| | |
|------------------------|------------|
| 5.2.3 字典管理..... | 71 |
| 5.2.4 参数管理..... | 73 |
| 5.2.5 用户管理..... | 74 |
| 5.2.6 基础数据..... | 76 |
| 5.2.7 工具管理..... | 78 |
| 5.3 发票购买模块..... | 81 |
| 5.3.1 发票购买..... | 81 |
| 5.3.2 发票审核..... | 83 |
| 5.3.3 发票分发..... | 84 |
| 5.3.4 发票核对..... | 86 |
| 5.3.5 发票查询..... | 88 |
| 5.4 发票领用模块..... | 89 |
| 5.4.1 领用申请..... | 89 |
| 5.4.2 领用审核..... | 91 |
| 5.4.3 发票库存..... | 92 |
| 5.5 发票开票模块..... | 93 |
| 5.5.1 开票操作..... | 93 |
| 5.5.2 发票签收..... | 97 |
| 5.5.3 发票验旧..... | 99 |
| 5.6 本章小结..... | 100 |
| 第六章 系统测试..... | 101 |
| 6.1 测试技术..... | 101 |
| 6.2 测试环境..... | 101 |
| 6.3 测试要求..... | 102 |
| 6.4 测试用例..... | 102 |
| 6.5 测试报告..... | 114 |
| 6.6 本章小结..... | 115 |
| 第七章 总结与展望..... | 116 |
| 7.1 总结..... | 116 |
| 7.2 展望..... | 116 |
| 参考文献..... | 118 |
| 致 谢..... | 119 |

Contents

| | |
|---|----------|
| Chapter 1 Introduction..... | 1 |
| 1.1 Background and Significance of the Research | 1 |
| 1.2 Current Research Status | 1 |
| 1.3 Main Research Contents..... | 2 |
| 1.4 Structure of the Dissertation | 2 |
| Chapter 2 Relate technology introduction | 3 |
| 2.1 Software Architecture Technology | 3 |
| 2.2 Object Oriented Design Technology | 4 |
| 2.3 Software Development Technology | 4 |
| 2.4 JAVA Technology | 5 |
| 2.5 SQL SERVER 2008 Technology | 7 |
| 2.6 Summary | 7 |
| Chapter 3 System Requirements Analysis | 8 |
| 3.1 System overview | 8 |
| 3.1.1 Planning Target | 8 |
| 3.1.2 Definition of terms | 8 |
| 3.2 Service description | 9 |
| 3.2.1 System Stakeholders | 9 |
| 3.2.2 System Boundary | 10 |
| 3.2.3 Business Process Analysis | 11 |
| 3.2.4 Business Form | 13 |
| 3.3 Data Description | 14 |
| 3.3.1 Static Data..... | 14 |
| 3.3.2 Dynamic Data | 15 |
| 3.3.3 Database Introduced | 16 |
| 3.4 Functional description | 17 |
| 3.4.1 Use Case Diagram..... | 17 |
| 3.4.2 Login Use Case | 18 |
| 3.4.3 Invoice Purchase Use Case | 20 |
| 3.4.4 Invoice Apply Use Case | 22 |
| 3.4.5 Invoice Bill Use Case | 24 |
| 3.4.6 Domain Model | 26 |
| 3.4.7 System Sequence Diagram | 26 |
| 3.5 Performance requirements | 28 |
| 3.5.1 Data accuracy | 28 |
| 3.5.2 Time characteristics | 29 |

| | |
|---|-----------|
| 3.5.3 Adaptability | 29 |
| 3.6 System Running Configure | 29 |
| 3.6.1 Environmental requirements | 29 |
| 3.6.2 Hardware interface | 31 |
| 3.6.3 Software interface..... | 31 |
| 3.6.4 Fault treatment | 34 |
| 3.7 Other Requirements | 35 |
| 3.7.1 Security Strategy | 35 |
| 3.7.2 Compatibility Requirements | 37 |
| 3.7.3 Usability Requirements | 37 |
| 3.7.4 Maintainability Requirements | 37 |
| 3.8 Summary..... | 37 |
| Chapter 4 The Overall Design Of System | 38 |
| 4.1 System Architecture Design | 38 |
| 4.2 System Module Design | 40 |
| 4.3 Database Design | 42 |
| 4.3.1 Design Principle | 42 |
| 4.3.2 Base Table Design | 43 |
| 4.3.3 System Table Design | 49 |
| 4.3.4 Coding Table Design | 49 |
| 4.3.5 Business Table Design | 50 |
| 4.4 E-R Diagram | 62 |
| 4.5 This Chapter Details | 64 |
| Chapter 5 System Design And Implementation | 65 |
| 5.1 Configuration Management | 65 |
| 5.1.1 Development Management | 65 |
| 5.1.2 Development Tool | 66 |
| 5.1.3 Development Environment | 67 |
| 5.2 System Management Related Model | 68 |
| 5.2.1 Login Management | 68 |
| 5.2.2 Home-Page Management | 70 |
| 5.2.3 Data Dictionary Management | 71 |
| 5.2.4 Parameter Management | 73 |
| 5.2.5 User Management | 74 |
| 5.2.6 Base Data | 76 |
| 5.2.7 Tools Management | 78 |
| 5.3 Invoice Purchase Module | 81 |

| | |
|--|------------|
| 5.3.1 Invoice Purchase..... | 81 |
| 5.3.2 Invoice Check | 83 |
| 5.3.3 Distribution Of Invoice..... | 84 |
| 5.3.4 Invoice Verification | 86 |
| 5.3.5 Invoice Inquiry | 88 |
| 5.4 Invoice Collar Model | 89 |
| 5.4.1 Receiving Application..... | 89 |
| 5.4.2 Using Audit | 91 |
| 5.4.3 Invoice Stock | 92 |
| 5.5 Invoice Billing Model | 93 |
| 5.5.1 Invoice Billing..... | 93 |
| 5.5.2 Invoice Receipt | 97 |
| 5.5.3 Invoice Expiration Checking | 99 |
| 5.6 Summary..... | 100 |
| Chapter 6 System Test | 101 |
| 6.1 Test Technology | 101 |
| 6.2 Test Environment | 101 |
| 6.3 Test Requirement | 102 |
| 6.4 Test Case..... | 102 |
| 6.5 Test Report | 114 |
| 6.6 This Chapter Details | 115 |
| Chapter 7 Conclusions and Future Work | 116 |
| 7.1 Conclusions | 116 |
| 7.2 Future Work | 116 |
| References..... | 118 |
| Acknowledgements | 119 |

第一章 绪论

1.1 研究背景及意义

银行业在中国，一直定位于特殊服务行业，有关银行业收费服务是否需要开票，一直存在诸多争论。按国家颁布的《发票管理办法》、《税收征收管理法》、《消费者权益保护法》等法规，银行业作为服务提供方，应该开具正式发票。但在实际执行中，不提供发票已成为行业惯例，其开具的交易凭条具有准发票的作用。另一方面，很多储户在银行办理收费业务时没有索要发票，而绝大多数企事业单位也都把银行开具的业务收费凭证、收据用以入账或报销，无意中也为银行不开正规发票提供了“正当”理由。随着中国银行业的快速发展，经营模式及业务种类越来越多元化，诸如黄金、白银、金融衍生物的交易量日渐庞大，交易纠纷越来越多，税收监管难度加大。从 2010 年起，国家新的《发票管理办法》^[1]出台有关银行业发票管理的规定，按照国家税务总局发票综合改革的要求，银行业代收费业务必须要开具发票，可以采用自开票系统，但要实现与各地税务机关网上开票系统的对接，以达到发票开具信息的适时传输。现实情况显示，大多数银行目前对发票还停留在手工管理模式，巨大的开票量导致出现诸如：分行与网点间信息交互不及时、客户获取发票周期长、网点发票保管及开具无法监控、不利于开票统计等一系列的问题。因此，开发银行业的发票管理软件，作为信息化的落地工具，就成为当前银行业的一个重要课题，税务机构也鼓励银行自主开发自开票系统，以提高发票管理水平。

1.2 研究目标

根据银行业发票管理的现状及业务流程，以信息化为手段，建立一个高效的银行业发票管理系统，全面管控从发票购买→发票分发→发票领用→发票审核→发票填报→发票打印→发票签收→发票验旧的完整过程，并可与税务系统数据对接，起到利于监管、方便统计的作用。

系统在 SOA 面向服务的架构下，采用 J2EE 技术、SQL Server2008 数据库和 Struts、

Spring、等开源框架，运用了大量先进的诸如 Ajax、RSS 等 web2.0 的技术，增强了系统的灵活性和开放性。

1.3 研究内容

本文介绍了研究背景、研究目标和业务需求，着重论述系统功能的设计和实现，从系统目标、设计思路和需求分析着手，描述系统总体架构的设计方案、系统设计的关键点和关键技术、系统的功能和数据库设计，并完成了系统的主要功能模块。具体研究内容如下：

- 1、需求分析：对银行业发票管理的业务流程进行梳理，分析各层级的角色需要完成的业务。从管理的角度思考哪些首段可以提升管理水平和辅助决策，整理一套适合国有、商业银行发票电子化管理的操作流程。
- 2、系统应用架构设计：根据系统需求分析，设计系统应用层次架构。
- 3、系统总体技术架构设计：依据需求设计通用性、可扩展性的系统技术架构。
- 4、设计并实现一个“银行业税务发票管理系统”。

1.4 论文的结构安排

本文共分为七章，各章节内容安排如下。

第一章，对当前银行业税务发票管理的现状进行分析，阐明课题的研究意义与主要研究内容。

第二章，介绍在银行业税务发票管理系统中应用到的一些关键技术。

第三章，详细分析银行业税务发票管理的业务流程、存在的问题和系统需求。

第四章，详细描述银行业税务发票管理系统的总体设计与数据库设计。

第五章，详细描述系统各个模块的实现过程。

第六章，描述系统的测试过程。

第七章，总结和展望，对项目的主要工作和论文的主要内容进行总结，并对人银行业税务发票管理系统的进一步研究进行展望。

第二章 相关技术介绍

2.1 软件架构技术

银行业普遍对软件系统的安全性、稳定性有较高要求，需要在软件开发的整个生命周期过程中进行全程质量管控。因此，采用“4+1”的软件体系架构模式^[2]进行开发。

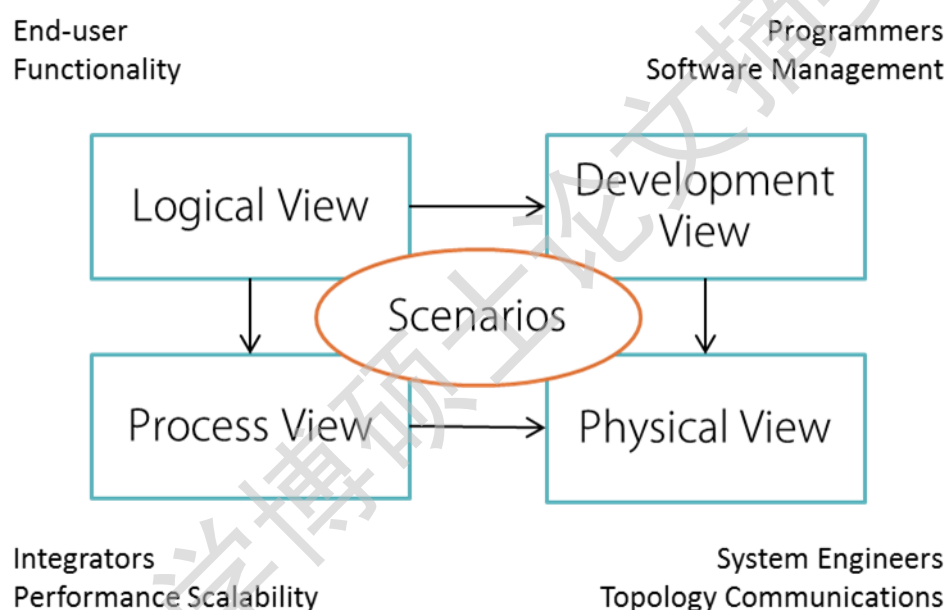


图 2-1 4+1 软件体系架构模式图

本模式使用多个并发的视图来组织软件架构的描述，每个视图仅用来描述一个特定的所关注的方面的问题集合。

- 逻辑视图（Logical View），设计的对象模型（使用面向对象的设计方法时）。
- 过程视图（Process View），捕捉设计的并发和同步特征。
- 物理视图（Physical View），描述了软件到硬件的映射，反映了分布式特性。
- 开发视图（Development View），描述了在开发环境中软件的静态组织结构。

架构视图是对于从某一视角或某一点上看到的系统所做的简化描述，描述中涵盖了系统的某一特定方面，而省略了与此方面无关的实体。架构视图如同在建筑学中的不同

种类的蓝图。使用多个并发的视图来组织软件架构的描述，每个视图仅用来描述一个特定的所关注的方面的问题集合。软件工程师使用单张视图来捕捉所有的系统架构要点，努力地单一视图中表达超过其表达限度的蓝图。

2.2 面向对象设计技术

面向对象是一种新兴的程序设计方法, 或者是一种新的程序设计规范(paradigm), 其基本思想是使用对象、类、继承、封装、消息等基本概念来进行程序设计。从现实世界中客观存在的事物(即对象)出发来构造软件系统, 并且在系统构造中尽可能运用人类的自然思维方式。力图使对计算机语言中对事物的描述与现实世界中该事物的本来面目尽可能的一致。开发一个软件是为了解决某些问题, 这些问题所涉及的业务范围称作该软件的问题域。其应用领域不仅仅是软件, 还有计算机体系结构和人工智能等^[3]。

2.3 软件开发技术

敏捷开发模式是一种从九十年代开始逐渐引起广泛关注的一些新型软件开发方法, 是一种应对快速变化的需求的一种软件开发能力。它们的具体名称、理念、过程、术语都不尽相同, 相对于“非敏捷”, 更强调程序员团队与业务专家之间的紧密协作、面对面的沟通(认为比书面的文档更有效)、频繁交付新的软件版本、紧凑而自我组织型的团队、能够很好地适应需求变化的代码编写和团队组织方法, 也更注重做为软件开发中人的作用^[4]。

在敏捷方法其独特之处以外, 他和其他的方法也有很多共同之处, 比如迭代开发, 关注互动沟通, 减少中介过程的无谓资源消耗。通常可以在以下方面衡量敏捷方法的适用性: 从产品角度看, 敏捷方法适用于需求萌动并且快速改变的情况, 如系统有比较高的关键性、可靠性、安全性方面的要求, 则可能不完全适合; 从组织结构的角度看, 组织结构的文化、人员、沟通则决定了敏捷方法是否适用。这些相关联的关键成功因素有:

组织文化必须支持谈判人员彼此信任, 人少但是精干, 开发人员所作决定得到认可,

环境设施满足成员间快速沟通之需要。最重要的因素恐怕是项目的规模。规模增长，面对面的沟通就愈加困难，因此敏捷方法更适用于较小的队伍，20、40 人或者更少。大规模的敏捷软件开发尚处于积极研究的阶段。

另外的问题是项目初期的大量设想或快速的需求收集可能导致项目走入误区，特别是客户对其自身需要毫无概念的情况下。与之类似，人之天性很容易造成某个人成为主导并将项目目标和设计引入错误方向的境况。开发者经常会把不恰当的方案授予客户，而直到最后出问题前都能获得客户认同。虽然理论上快速交互的过程可以限制这些错误的发生，但前提是有效的负反馈，否则错误会迅速膨胀。

敏捷方法有时候被误认为是无计划性和纪律性的方法，实际上更确切的说法是敏捷方法强调适应性而非预见性。适应性的方法集中在快速适应现实的变化。当项目的需求起了变化，团队应该迅速适应。这个团队可能很难确切描述未来将会如何变化。

相比迭代式开发两者都强调在较短的开发周期提交软件，敏捷开发的周期可能更短，并且更加强调队伍中的高度协作。

2.4 JAVA 技术

Java 是一种可以撰写跨平台应用程序的面向对象的程序设计语言，是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台（即 J2SE, J2EE, J2ME）的总称。Java 技术具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于个人 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网，同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下，Java 更具备了显著优势和广阔前景^[5]。

其中 J2EE 核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格，让各种依循 J2EE 架构的不同平台之间，存在良好的兼容性，解决过去企业后端使用的信息产品彼此之间无法兼容，企业内部或外部难以互通的窘境。在 J2EE 技术规范下，具体应用以下相关软件技术：

● Struts 技术

Struts 是开源软件。使用 Struts 的目的是为了帮助我们减少在运用 MVC 设计模型来开发 Web 应用的时间。struts 框架具有组件的模块化，灵活性和重用性的优点，同

Degree papers are in the "[Xiamen University Electronic Theses and Dissertations Database](#)". Full texts are available in the following ways:

1. If your library is a CALIS member libraries, please log on <http://etd.calis.edu.cn/> and submit requests online, or consult the interlibrary loan department in your library.
2. For users of non-CALIS member libraries, please mail to etd@xmu.edu.cn for delivery details.

厦门大学博硕士论文摘要库